TEETH DETECTION AND DENTAL PROBLEM CLASSIFICATION IN PANORAMIC X-RAY IMAGES USING DEEP LEARNING AND IMAGE PROCESSING TECHNIQUES

A Mini Project Report

submitted by

Namitha C (MES20MCA-2036)

to

the APJ Abdul Kalam Technological University
in partial fulfillment of the requirements for the award of the Degree

of

Master of Computer Applications

Department of Computer Applications

MES College of Engineering
Kuttippuram, Malappuram - 679 582

February 2022

DECLARATION

I undersigned hereby declare that the project report **Teeth detection and Dental problem classification in panoramic X-ray images using deep learning and image processing techniques**, submitted for partial fulfillment of the requirements for the award of degree of Master of Computer Applications of the APJ Abdul Kalam Technological University, Kerala, is a bona fide work done by me under supervision of Mr.Muhammad Jabir C,Assistant Professor,Department of Computer Applications. This submission represents my ideas in my own words and where ideas or words of others have been included, I have adequately and accurately cited and referenced the original sources. I also declare that I have adhered to ethics of academic honesty and integrity and have not misrepresented or fabricated any data or idea or fact or source in my submission. I understand that any violation of the above will be a cause for disciplinary action by the institute and/or the University and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been obtained. This report has not been previously formed the basis for the award of any degree, diploma or similar title of any other University.

Place: KUTTIPPURAM

Date:28/2/2022

Namitha C (MES20MCA-2036)

DEPARTMENT OF COMPUTER APPLICATIONS
MES COLLEGE OF ENGINEERING, KUTTIPPURAM

CERTIFICATE

This is to certify that the report entitled **Teeth detection and Dental problem classification in panoramic X-ray images using deep learning and image processing techniques** is a bona fide record of the Mini Project work carried out by **Namitha C (MES20MCA-2036)** submitted to the APJ Abdul Kalam Technological University, in partial fulfillment of the requirements for the award of the Master of Computer Applications, under my guidance and supervision. This report in any form has not been submitted to any other University or Institution for any purpose.

Internal Supervisor(s)

External Supervisor(s)

Head Of The Department

Acknowledgements

My endeavor stands incomplete without dedicating my gratitude to a few people who have contributed towards the successful completion of my project.I pay my gratitude to the Almighty for his invisible help and blessing for the fulfillment of this work.At the outset I express my heartful thanks to my guide Mr. Muhammad Jabir C for his valuable guidance and supervision. I take this opportunity to express my profound gratitude to Ms.Priya J D as my project coordinator for her valuable support, timely advise and strict schedules to complete my project. I also grateful to all my teaching and non-teaching staff for their encouragement, guidance and whole-hearted support. Last but not least, I gratefully indebted to my family and friends, who gave me a precious help in doing our project.

Namitha C (MES20MCA-2036)

Abstract

Abstract— Deep convolutional neural networks, have gained a lot popularity in medical research due to their impressive results in detection, prediction and classification. Analysis of panoramic dental radiographies help specialists observe problems in poor visibility areas, inside the buccal cavity or in hard to reach areas. However, poor image quality or fatigue can cause the diagnosis to vary, which can ultimately hinder the treatment. In this paper we propose a novel approach of automatic teeth detection and dental problem classification using panoramic X-Ray images which can aid the medical staff in making decisions regarding the correct diagnosis. For this endeavor panoramic radiographies were collected from three dental clinics and annotated, highlighting 14 different dental issues that can appear. A CNN was trained using the annotated data for obtaining semantic segmentation information. Next, multiple image processing operations were performed for segmenting and refining the bounding boxes corresponding to the teeth detections. Finally, each tooth instance was labeled and the problem affecting it was identified using a histogram-based majority voting within the detected region of interest. The implemented solution was evaluated with respect to several metrics like intersection over union for the semantic segmentation and accuracy, precision, recall and F1-score for the generated bounding box detections. The results were compared qualitatively with the data obtained from other approaches illustrating the superiority of the proposed solution.

Keywords:CNN; Medical Imaging; Image Processing; Machine Learning; Dental Informatics; X-Ray Images; Semantic Segmentation;

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Background

In this paper we propose a novel approach of automatic teeth detection and dental problem prediction using panoramic X-Ray images which can aid the medical staff in making decisions regarding the correct diagnosis. The image processing techniques include Semantic segmentation is process of assigning a label to every pixel in the image for complete scene understanding, the algorithm should figure out the object and it's pixels. The conversion of colour image into grey scale image is converting the RGB values into grey scale value. CNN extract the feature of image and convert it into lower dimension without loosing its characteristics.

1.1.1 Motivation

Nowadays dental radiography shows the entire mouth area where all the teeth can be seen. It shows complete images of the jaws and the skull thus giving the dentist an overview about the patient's problems. Dental radiography is used by dentists to observe problems in hard-to-reach areas or with a poor visibility inside the buccal cavity. The interpretation of the radiography is done manually by the dentist, who identifies each tooth and the existing problem where appropriate. However, if the X-ray radiography is not clear it can cause problems when analysed and thus lead to misinterpretation. So it's essential to have a system to predict the problems.

1.2 Objective

Main objective of the project is to help the patients to check the issues in their teeth and take proper treatment regarding the problems.Choose better tips for maintaining the teeth in proper way.Getting chance to choose hospitals by refering the feedback from different users.

1.3 Contribution

Users will be getting the predicted results within few seconds and will get a chance to choose hospitals within their location.

1.4 Report Organization

The project report is divided into four sections. Section 2 describes literature survey. Section 3 describes the methodology used for implementing the project. Section 4 gives the results and discussions. Finally Section 5 gives the conclusion.

Chapter 2

Literature Survey

D.V. Tuzoff, L.N. Tuzova, M.M. Bornstein, A.S. Krasnov, M.A. Kharchenko, S.I. Nikolenko, M.M. Sveshnikov, G.B. Bednenko has proposed analysis of dental radiographs is an important part of the diagnostic process in daily clinical practice. Interpretation by an expert includes teeth detection and numbering. In this project, a novel solution based on convolutional neural networks (CNNs) is proposed that performs this task automatically for panoramic radiographs.

E.A. Mendonçahas proposed clinical decision-support systems (CDSSs) are computer programs that are designed to provide expert support for health professionals making clinical decisions. The goal of these systems is to help health professionals analyze patient data and make decisions regarding diagnosis, prevention, and treatment of health problems. This article discusses the characteristics of such systems, addresses the challenges in developing them, identifies potential barriers for their use in clinical practice, and provides perspectives for the future.

Chapter 3

Methodology

3.1 Introduction

In recent years, medical imaging technologies such as computed tomography (CT) or X-rays have aided the treatment and diagnosis of different diseases. In the field of dentistry, dental informatics is an emergent field, which, in addition to helping improve the treatment and diagnosis process, saves time and reduces stress during the daily routine. The use of highresolution imaging sensors and biosensors has led to the generation of massive amounts of data, which can be interpreted by computer programs that can help dental professionals in making decisions related to prevention, diagnosis or treatment planning, among others. Radiographies are obtained by the passage of X-rays, produced by an X-Ray generator, through the oral cavity. Radiation can be absorbed by some tissues, or it can pass through the patient being absorbed by a detector. This process is called projective radiography and it generates twodimensional images which represents internal structures of the human body. Dental radiographies can be classified in two categories: intraoral where the film is positioned inside the buccal cavity, and extraoral where the patient is positioned between the source that emanates X-rays and the radiographic film. Most common types of dental X-rays are bitewing, periapical, which are intraoral, and panoramic which is extraoral. A panoramic dental radiography shows the entire mouth area where all the teeth can be seen. It also shows the jaws and the skull thus giving the dentist an overview about the patient's problems. The panoramic dental radiography is used by dentists to observe problems in hard-to-reach areas or with a poor visibility inside the buccal cavity.

The interpretation of the radiography is done manually by the dentist, who identifies each tooth and the existing problem where appropriate. However, if the X-ray radiography is not clear it can cause problems when analyzed and thus lead to misinterpretation. Convolutional Neural Networks (CNN) are the preferred solutions for medical imaging analysis, and have been employed in many clinical fields.

Some successful applications in which CNNs have been applied include the assessment of breast cancer in mammograms , skin cancer in clinical skin screenings or recognition of hepatocellular carcinoma areas from ultrasound images. In the field of dentistry CNNs have been applied to detect periodontal bone loss , carries apical lesions. CNNs can also be used to detect different structures, classify them and segment them. When using supervised learning, Neural Networks need to be trained and optimized on an image database in order to obtain an accurate result.We manually annotate panoramic radiographies in order to train the semantic segmentation CNN

• Segment semantically the panoramic X-Ray image, for 15 semantic classes depicting different dental problems, using a CNN Detect and label each tooth or group of teeth (depending on the scenario) and the dental problem affecting it using multiple image processing techniques

• Implement a refinement method, in order to eliminate small inconsistencies

• Evaluate and compare the proposed solution, with other CNNs created for the same task.

This project was developed using Agile Developement Model.The entire project was divided into three sprints.In the first sprint table design,form design and basic coding completed.In second sprint creation of dataset,pre-processing of x-ray images and prediction completed.In the third sprint User prediction process, Testing data,Output generation .

3.2 Modules

ADMIN

Login

View users

Add tips

View feedback

Add and manage dataset

Add hospitals

USER

Register

Login

View tips

Feedback

Upload X-Ray image

Upload X-Ray image and predict problem

View nearest hospital

3.3 CNN

There are multiple convolutional layers extracting features from the image and finally the output layer. CNN extracts the feature of image and convert it into lower dimensions without loosing its characteristics. The role of cnn is to reduce the images into a form which is easier to process,without losing features which are critical for getting good prediction. Each input layer is connected to the next hidden layer. The output obtained is compared with dataset and then predict the result.Segmenting teeth from the X-ray film and performing numbering for each tooth, the testing teeth can be compared only with those having the same numbers in the database, thus the computational efficiency and accuracy can be improved. Further, the oral medical resources are sparse in several developing countries. Dentists usually need to serve numerous patients every day. As an important auxiliary diagnostic tool, a large number of dental X-ray films are photographed daily. A Convolutional Neural Network (ConvNet/CNN) is a Deep Learning algorithm which can take in an input image, assign importance (learnable weights and biases) to various aspects/objects in the image and be able to differentiate one from the other. The pre-processing required in a ConvNet is much lower as compared to other classification algorithms. While in primitive methods filters are hand-engineered, with enough training, ConvNets have the ability to learn these filters/characteristics.

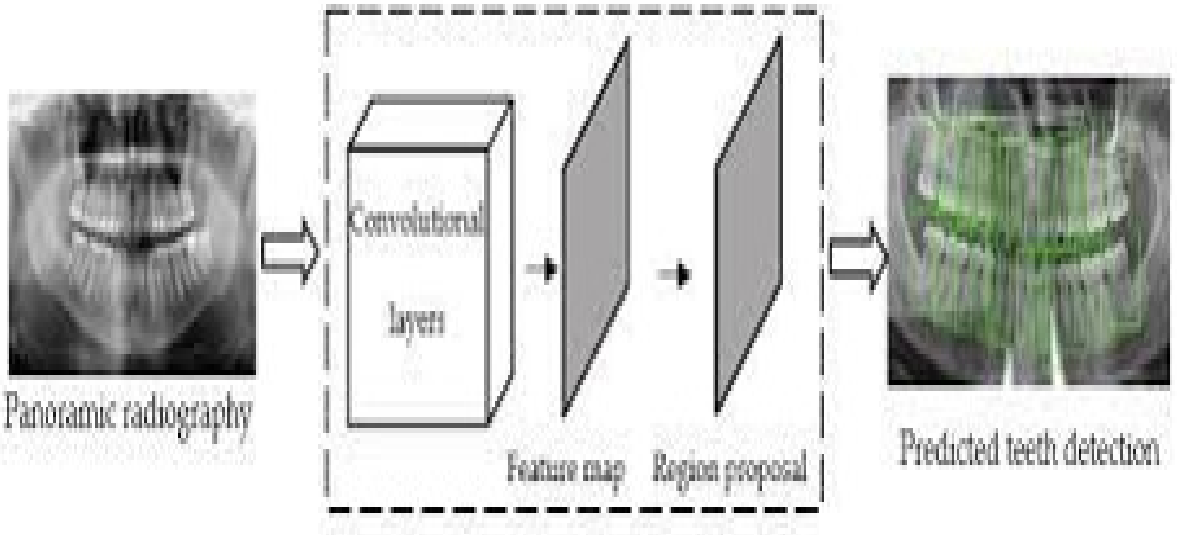

Figure 3.1: cnn levels

3.4 User Story

A key component of agile software development is putting people first, and user-stories put actual end users at the center of the conversation. Stories use non-technical language to provide context for the development team and their efforts. After reading a user story, the team knows why they are building what they're building and what value it creates. A user story is a tool used in agile software development to capture a description of a software feature from an enduser perspective. The user story describes the type of user, what they want and why. A user story helps to create a simplified description of a requirement. User stories are one of the core components of an agile program. They help provide a user-focused framework for daily work which drives collaboration, creativity, and a better product overall. The user story of system is given in Table 3.2

UserStoryID	As a <type of user>	I want to	So that I can
1	Admin	login	login successful with correct username and password
2	Admin	Manage hospitals	Add,view,edit&delete hospitals
3	Admin	View users	View all registered users
4	Admin	Add tips	Add tips to help users
5	Admin	View feedback	View feedback about doctor from users
6	User	Login	login successful with correct username and password
7	User	View tips	View all tips from admin
8	User	Feedback	Provide feedback to admin
9	User	Upload image	uploads the image and views the result
10	User	View nearest Hospital	Views nearest hospital for consulting doctor

Figure 3.2: user story

3.5 Product backlog

A product backlog is a list of the new features, changes to existing features, bug fixes, infrastructure changes or other activities that a team may deliver in order to achieve a specific outcome.The product backlog is the single authoritative source for things that a team works on. That means that nothing gets done that isn't on the product backlog. Conversely, the presence of a product backlog item on a product backlog does not guarantee that it will be delivered. It represents an option the team has for delivering a specific outcome rather than a commitment.It should be cheap and fast to add a product backlog item to the product backlog, and it should be equally as easy to remove a product backlog item that does not result in direct progress to achieving the desired outcome or enable progress toward the outcome. The Scrum Product Backlog is simply a list of all things that needs to be done within the project. It replaces the traditional requirements specification artifacts. These items can have a technical nature or can be user-centric e.g. in the form of user stories.The product backlog of the system is given in Table 3.3

User Story ID	Priority <High/Medium/Low>	Size (Hours)	Sprint <#>	Status <Planned/In progress/Completed>	Release Date	Release Goal
1	Medium	2	1	Completed		Table design
2	High	3		Completed		Form design
3	High	5		Completed		Basic coding
3	Medium	5	2	Completed		Creation of dataset
4	High	5		Completed		Pre-processing x-ray images
5	High	5	3	Completed		prediction
6	high	5		Completed		User prediction process.
7	Medium	5	4	Completed		Testing data
8	High	5		Completed		Output generation

Figure 3.3: product backlog

3.6 Project plan

A project plan that has a series of tasks laid out for the entire project, listing task durations, responsibility assignments, and dependencies. Plans are developed in this manner based on the assumption that the Project Manager, hopefully along with the team, can predict up front everything that will need to happen in the project, how long it will take, and who will be able to do it. Project paln is given in Table 3.4

User Story ID	Task Name	Start Date	End Date	Days	Status
1	Sprint 1	26/12/2021	28/12/2021	2	Completed
2		29/12/2021	31/12/2021	3	Completed
3		03/12/2021	08/01/2022	5	Completed
4	Sprint 2	09/01/2022	16/01/2022	8	Completed
5		18/01/2022	22/01/2022	5	Completed
6	Sprint 3	23/01/2022	27/01/2022	5	Completed
7		30/01/2022	05/02/2022	7	Completed
8	Sprint 4	06/02/2022	10/02/2022	5	Completed
9		16/02/2022	20/02/2022	4	Completed

Figure 3.4: project plan

3.7 Sprint backlog plan

The sprint backlog is a list of tasks identified by the Scrum team to be completed during the Scrum sprint. During the sprint planning meeting, the team selects some number of product backlog items, usually in the form of user stories, and identifies the tasks necessary to complete each user story. Most teams also estimate how many hours each task will take someone on the team to complete.

Backlog Item	Status & completion date	Original estimate in hours	Day1	Day2	Day3	Day4	Day5	Day6	Day7	Day8	Day9	Day10	Day11	Day12	Day13	Day14
User story #1,#2, #3,#4		hrs	Hrs	hrs	hrs	hrs	hrs	hrs	hrs	hrs	hrs	hrs	hrs	hrs	hrs	hrs
Table design	28/12/2021	2	1	1	0	0	0	0	0	0	0	0	0	0	0	0
Form design	31/12/2021	3	0		1	1	1	0	0	0	0	0	0	0	0	0
Coding	08/01/2021	5	0	0	0	0	0	1	1	1	1	1	0	0	0	0
User story #5,#6, #7,#8																
Creation of dataset	16/01/2022	5	1	1	0	1	1	1	0	1	0	0	0	0	0	0
Pre-processing x-ray images	22/01/2022	5	0	0	0	0	0	0	0	1	1	0	1	1	1	0
prediction	27/01/2022	5	1	1	1	0	1	1	0	0	0	0	0	0	0	0
User prediction process	05/02/2022	5	0	0	0	0	0	0	0	1	1	1	1	1	0	0
Testing data	10/01/2022	5	1	1	1	1	1	0	0	0	0	0	0	0	0	0
User story #9																
Output generation	20/02/2022	5	0	0	0	0	0	0	2	2	2	0	0	0	0	0
Totall		40	4	4	3	3	4	3	3	5	4	2	2	2	1	0

Figure 3.5: sprint backlog

3.8 Sprint actual

Actual sprint backlog is what adequate sprint planning is actually done by project team there may or may not be difference in planned sprint backlog. The detailed sprint backlog (Actual) is given below

Backlog item	Status and completion date	Original estimate in hours	Day1	Day2	Day3	Day4	Day5	Day6	Day7	Day8	Day9	Day10	Day11	Day12	Day13	Day14
User story#1,#2,#3			hrs	hrs	hrs	hrs	hrs	hrs	hrs	hrs	hrs	hrs	hrs	hrs	hrs	hrs
Table design	28/12/2021	2	1	1	0	0	0	0	0	0	0	0	0	0	0	0
Form design	31/12/2021	3	0	0	2	1	0	0	0	0	0	0	0	0	0	0
Basic coding	08/01/2022	5	0	0	0	0	0	1	1	1	2	0	0	0	0	0
User story #4,#5																
collects the features of the disease	22/01/2022	8	2	0	0	2	0	2	0	0	1	0	1	0	0	0
Training the data	22/01/2022	5	1	0	0	0	2	0	0	1	0	0	1	0	0	0
User story #6,#7																
classify different images using CNN	5/02/2022	5	1	0	0	0	2	0	0	0	0	2	0	0	0	0
find issue or not	17/02/2022	7	2	0	0	0	0	2	0	0	0	0	2	1	0	0
User story #8,#9																
Testing data	20/02/2022	5	2	0	0	0	0	0	0	2	0	0	0	0	1	0
Output generation	20/02/2022	4	0	0	0	0	0	0	2	2	0	0	0	0	0	0
Total		44	9	1	2	3	4	5	3	6	3	2	4	1	1	0

Figure 3.6: sprint actual

Chapter 4

Results and Discussions

4.1 Datasets

Dataset from kaggle website used to analyse the images of teeth having issues and no issues.

4.2 Results

Project uses images from dataset and trained it using cnn algorithm.From dataset teeth having issues and no issues are taken and evaluated.After trainning the dataset,x-ray images uploaded by different users were tested by different image processing techniques and with the help of cnn algorithm.The images in dataset as well as x-ray images uploaded by users were matched and predict wheather teeth having any issue.

4.3 Screenshots

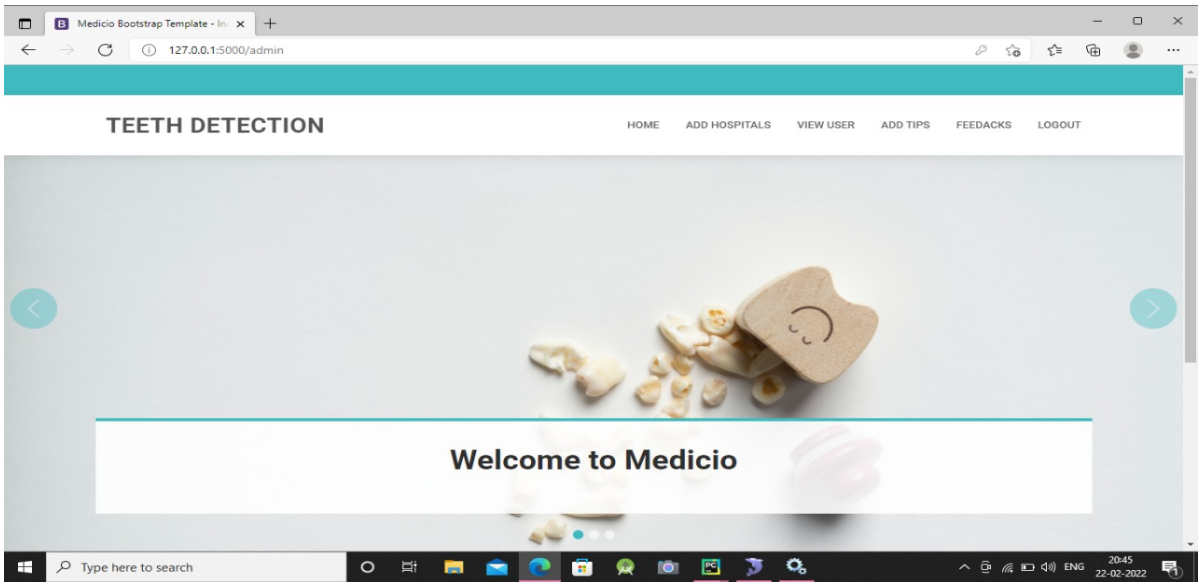

Figure 4.1: screenshots

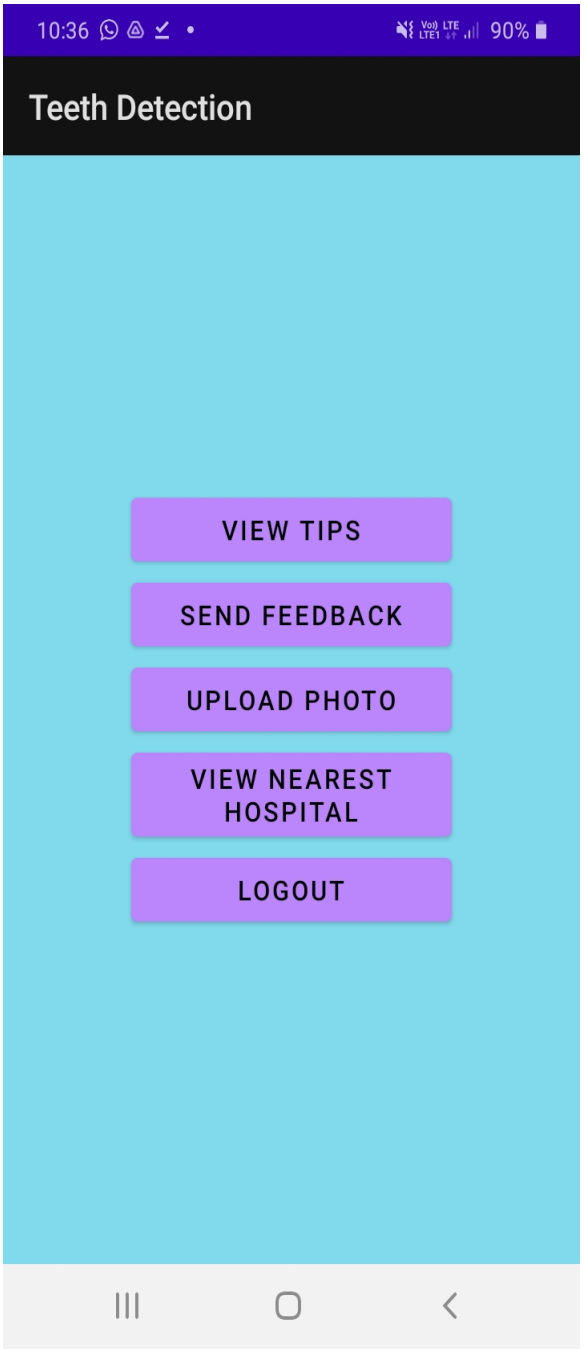

Figure 4.2: screenshots

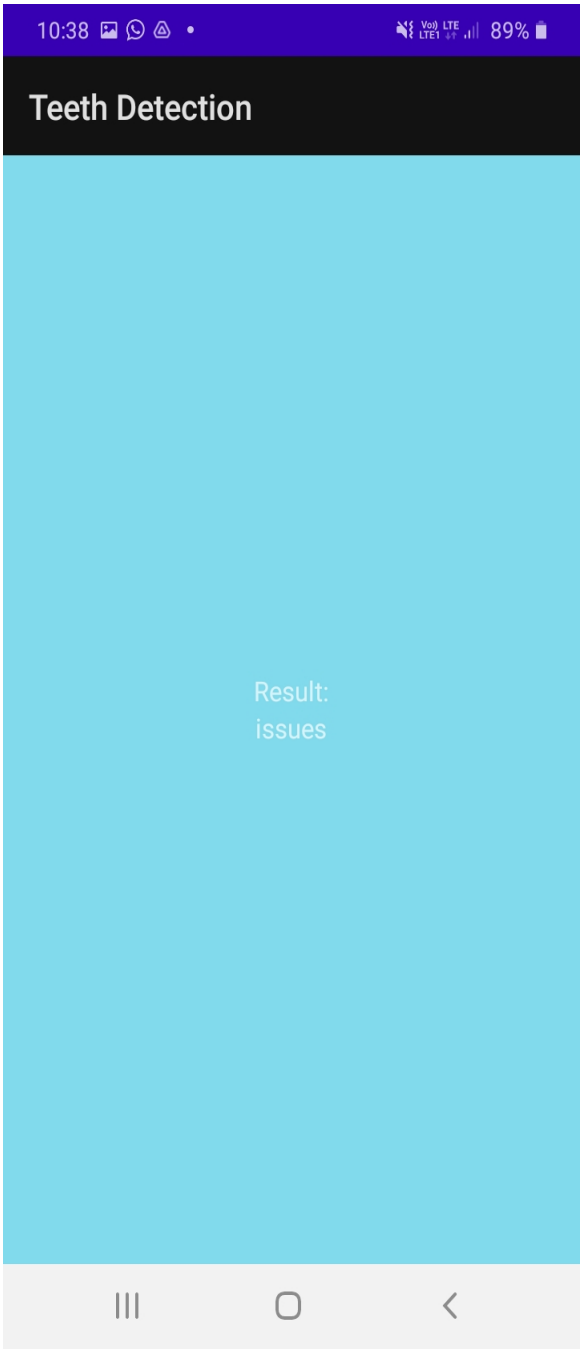

Figure 4.3: screenshots

Chapter 5

Conclusions

This paper presented a novel teeth detection and dental problem classification approach using panoramic dental radiographies. For achieving the desired results images were collected from three different dental clinics and were annotated at pixel level, highlighting 14 different problems that can affect teeth. The annotated data was augmented using multiple operations, and a semantic segmentation CNN was trained using these images. Afterwards, the semantic segmentation image was binarized using multiple thresholds and a two-step labeling algorithm was used to detect each tooth instance. The bounding boxes corresponding to each instance are determined, and a refinement algorithm is applied in order to remove the regions that resulted from inconsistences in the semantic segmentation image. The bounding boxes of each instance is projected onto the semantic segmentation image and a histogram-based majority voting operation is performed in order to find the main semantic class of each tooth, which corresponds to the dental problem affecting the tooth. Each tooth or group of teeth, depending on the scenario, are numbered and a report containing the dental problems for each instance is generated aiding the medical staff in the diagnosis process. The implemented solution is evaluated using multiple metrics and compared to similar algorithms.

References

[1] **D. Shen, G. Wu, H.-I. Suk** 'Deep Learning in Medical Image Analysis', Annual review of biomedical engineering, 19, pp. 221-248, 2017.

[2] **D.V. Tuzoff, L.N. Tuzova, M.M. Bornstein, A.S. Krasnov, M.A. Kharchenko, S.I. Nikolenko, M.M. Sveshnikov, G.B. Bednenko,** 'Tooth detection and numbering in panoramic radiographs using convolutional neural networks', Dentomaxillofacial Radiol, 48, 20180051, 2019.

[3] **E.A. Mendonça,**'Clinical decision support systems: Perspectives in dentistry', Journal of dental education, 68, pp. 589–597, 2004.

[4] **J.I. Haring, L. Jansen,** Jansen, Dental radiography: principles and techniques, Second edition, W.B. Saunders Company, 2000.

Appendix

Source Code

Android

```
  from flask import *
app=Flask(__name__)
from src.dbconnector import *
app.secret_key="qwerty"
import functools

def login_required(func):
   @functools.wraps(func)
   def secure_function():
      if "lid" not in session:
         return render_template('login.html')
      return func()
 return secure_function
@app.route('/logout')
def logout():
   session.clear()
   return redirect('/')
@app.route('/')
def log():
   return render_template("login.html")
@app.route('/login',methods=['post'])
def login():
   uname=request.form['textfield']
   passwd=request.form['textfield2']
   qry="SELECT * FROM `login` WHERE `username`=%s AND `password`=%s"
   val=(uname,passwd)
   res=selectone(qry,val)
   if res is None:
      return '''<script>alert('Invalid username or password');window.location='/'</script>'''
   elif res[3]=='admin':
      session['lid']=res[0]
      return redirect('/admin')
   else:
      return '''<script>alert('Invalid username or password');window.location='/'</script>'''
@app.route('/admin')
def admin():
   return render_template("adminhome.html")

@app.route('/addhosp',methods=['post'])
@login_required

def addhosp():
```

```
    return render_template("add_hospital.html")

@app.route('/add_hospital',methods=['post'])
@login_required
def add_hospital():
    hname=request.form['textfield']
    place = request.form['textfield2']
    post=request.form['textfield3']
    pin=request.form['textfield4']
    email=request.form['textfield5']
    contact=request.form['textfield6']
    latitude=request.form['textfield7']
    longitude=request.form['textfield8']
 qry="INSERT INTO `hospital` VALUES(NULL,%s,%s,%s,%s,%s,%s,%s,%s)"
    val=(hname,place,post,pin,email,contact,latitude,longitude)
    iud(qry,val)
    return '''<script>alert('Registred');window.location='/manage_hospital'</script>'''

@app.route('/delete_hospital')
@login_required
def delete_hospital():
    id=request.args.get('id')
    qry="DELETE FROM `hospital` WHERE `login_id`=%s"
    val=str(id)
    iud(qry,str(val))
    qry1="DELETE FROM `login` WHERE `id`=%s"
    val1=(str(id))
    iud(qry1,val1)
    return '''<script>alert('deleted');window.location='/manage_hospital'</script>'''
@app.route('/edit_hospital')
@login_required
def edit_hospital():
    id = request.args.get('id')
    session['hid']=id
    qry="SELECT * FROM `hospital` WHERE `hospital`.`login_id`=%s"
    res=selectone(qry,str(id))
    return render_template("edit_hospital.html",val=res)
@app.route('/update_hospital',methods=['post'])
@login_required
def update_hospital():
    hid=session['hid']
    hname = request.form['textfield']
    place = request.form['textfield2']
    post = request.form['textfield3']
    pin = request.form['textfield4']
    email = request.form['textfield5']
    contact = request.form['textfield6']
    latitude = request.form['textfield7']
    longitude = request.form['textfield8']
    qry="UPDATE `hospital` SET
          `hospital_name`=%s,`place`=%s,`post`=%s,`pin`=%s,`email`=%s,`contact`=%s,`latitude`=%s,`longitude`=%s WHERE
          `login_id`=%s"
    val=(hname,place,post,pin,email,contact,latitude,longitude,str(hid))
    iud(qry,val)
    return '''<script>alert('updated');window.location='/manage_hospital'</script>'''
@app.route('/add_tip',methods=['post'])
@login_required
def add_tip():
    return render_template("add_tip.html")


@app.route('/adding_tip',methods=['post'])
@login_required
```

Appendix

```
def adding_tip():
    sub=request.form['textfield']
    tip=request.form['textarea']
    qry="INSERT INTO `tip` VALUES(NULL,%s,%s)"
    val=(sub,tip)
    iud(qry,val)
    return redirect("manage_tip")
@app.route('/edit_tip')
@login_required
def edit_tip():
    id=request.args.get('id')
    session['tid']=id
    qry="select * from tip where tid=%s"
    res=selectone(qry,session['tid'])
 return render_template("edit_tip.html",val=res)
@app.route('/editing_tip',methods=['post'])
@login_required
def editing_tip():
    sub=request.form['textfield']
    tip=request.form['textarea']
    qry="update tip set subject=%s,tip=%s where tid=%s"
    val=(sub,tip,session['tid'])
    iud(qry,val)
    return redirect("manage_tip")
@app.route('/deletetip')
@login_required
def deletetip():
    id=request.args.get('id')
    qry="delete from tip where tid=%s"
    iud(qry,id)
    return redirect("manage_tip")
@app.route('/manage_hospital')
@login_required
def manage_hospital():
    qry="SELECT * FROM `hospital`"
    res=select(qry)
    return render_template("manage_hospital.html",val=res)
@app.route('/manage_tip')
@login_required
def manage_tip():
    qry="select * from tip"
    res=select(qry)
    return render_template("manage_tip.html",val=res)
@app.route('/view_feedback')
@login_required
def view_feedback():
    qry="SELECT `user`.`fname`,`lname`,`feedback`.* FROM `feedback` JOIN `user` ON `user`.`login_id`=`feedback`.`userid`"
    res=select(qry)
    return render_template("view_feedback.html",val=res)
@app.route('/view_user')
@login_required
def view_user():
    qry="SELECT * FROM `user`"
    res=select(qry)

    return render_template("view_user.html",val=res)
app.run(debug=True)

//webcode


from flask import *
app=Flask(__name__)
```

Appendix

```
from src.dbconnector import *
app.secret_key="qwerty"
@app.route('/')
def log():
   return render_template("login.html")
@app.route('/login',methods=['post'])
def login():
   uname=request.form['textfield']
   passwd=request.form['textfield2']
   qry="SELECT * FROM `login` WHERE `username`=%s AND `password`=%s"
   val=(uname,passwd)
   res=selectone(qry,val)
   if res is None:
      return '''<script>alert('Invalid username or password');window.location='/'</script>'''
   elif res[3]=='admin':
      return redirect('/admin')
   else:
      return '''<script>alert('Invalid username or password');window.location='/'</script>'''
@app.route('/admin')
def admin():
   return render_template("adminhome.html")
@app.route('/addhosp',methods=['post'])
def addhosp():
   return render_template("add_hospital.html")

@app.route('/add_hospital',methods=['post'])
def add_hospital():
   hname=request.form['textfield']
   place = request.form['textfield2']
   post=request.form['textfield3']
   pin=request.form['textfield4']
   email=request.form['textfield5']
   contact=request.form['textfield6']
   latitude=request.form['textfield7']
   longitude=request.form['textfield8'] uname=request.form['textfield9']
   passwd=request.form['textfield10']
 qry="INSERT INTO `login` VALUES(NULL,%s,%s,'hospital')"
   val=(uname,passwd)
   id=iud(qry,val)
qry="INSERT INTO `hospital` VALUES(NULL,%s,%s,%s,%s,%s,%s,%s,%s,%s)"
   val=(str(id),hname,place,post,pin,email,contact,latitude,longitude)
   iud(qry,val)
   return '''<script>alert('Registred');window.location='/admin'</script>'''
@app.route('/delete_hospital')
def delete_hospital():
   id=request.args.get('id')
   qry="DELETE FROM `hospital` WHERE `login_id`=%s"
   val=str(id)
   iud(qry,str(val))
   qry1="DELETE FROM `login` WHERE `id`=%s"
   val1=(str(id))
   iud(qry1,val1)
   return '''<script>alert('deleted');window.location='/manage_hospital'</script>'''
@app.route('/edit_hospital')
def edit_hospital():
   id = request.args.get('id')
   session['hid']=id
   qry="SELECT * FROM `hospital` WHERE `hospital`.`login_id`=%s"
   res=selectone(qry,str(id))
   return render_template("edit_hospital.html",val=res)
@app.route('/update_hospital',methods=['post'])
def update_hospital():
   hid=session['hid']
```

Appendix

```
    hname = request.form['textfield']
    place = request.form['textfield2']
    post = request.form['textfield3']
    pin = request.form['textfield4']
    email = request.form['textfield5']
    contact = request.form['textfield6']
    latitude = request.form['textfield7']
    longitude = request.form['textfield8']
qry="UPDATE `hospital` SET
       `hospital_name`=%s,`place`=%s,`post`=%s,`pin`=%s,`email`=%s,`contact`=%s,`lattitude`=%s,`longitude`=%s WHERE
       `login_id`=%s"
    val=(hname,place,post,pin,email,contact,latitude,longitude,str(hid))
    iud(qry,val)
 return '''<script>alert('updated');window.location='/manage_hospital'</script>'''
@app.route('/add_tip')
def add_tip():
    return render_template("add_tip.html")
@app.route('/edit_tip')
def edit_tip():
    return render_template("edit_tip.html")
@app.route('/manage_hospital')
def manage_hospital():
    qry="SELECT * FROM `hospital`"
    res=select(qry)
    return render_template("manage_hospital.html",val=res)
@app.route('/manage_tip')
def manage_tip():
    return render_template("manage_tip.html")
@app.route('/view_feedback')
def view_feedback():
    return render_template("view_feedback.html")
@app.route('/view_user')
def view_user():
    qry="SELECT * FROM `user`"
    res=select(qry)
return render_template("view_user.html",val=res)
app.run(debug=True)
```

Database Design

Attribute Name	Datatype	length	Description
id	Integer	11	Primary Key
username	Varchar	60	
type	Varchar	60	

Table A.1: login

Attribute Name	Datatype	length	Description
id	Integer	11	Primary Key
hospital name	Varchar	60	
place	varchar	60	
post	varchar	60	
pin	bigint	20	
email	varchar	60	
contact	bigint	20	
lattitude	varchar	60	
longitude	varchar	60	

Table A.2: hospital

Attribute Name	Datatype	length	Description
fid	Integer	11	Primary Key
userid	int	11	
feedback	Varchar	50	
date	Varchar	50	

Table A.3: feedback

Attribute Name	Datatype	length	Description
tid	Integer	11	Primary Key
subject	varchar	50	
tip	Varchar	50	

Table A.4: tip

Attribute Name	**Datatype**	**length**	**Description**
uid	Integer	11	Primary Key
login id	int	11	
fname	Varchar	60	
lname	Varchar	60	
gender	Varchar	60	
dob	Varchar	60	
place	Varchar	60	
post	Varchar	60	
pin	bigint	20	
email	varchar	60	
contact	bigint	20	

Table A.5: user

Dataflow Diagram

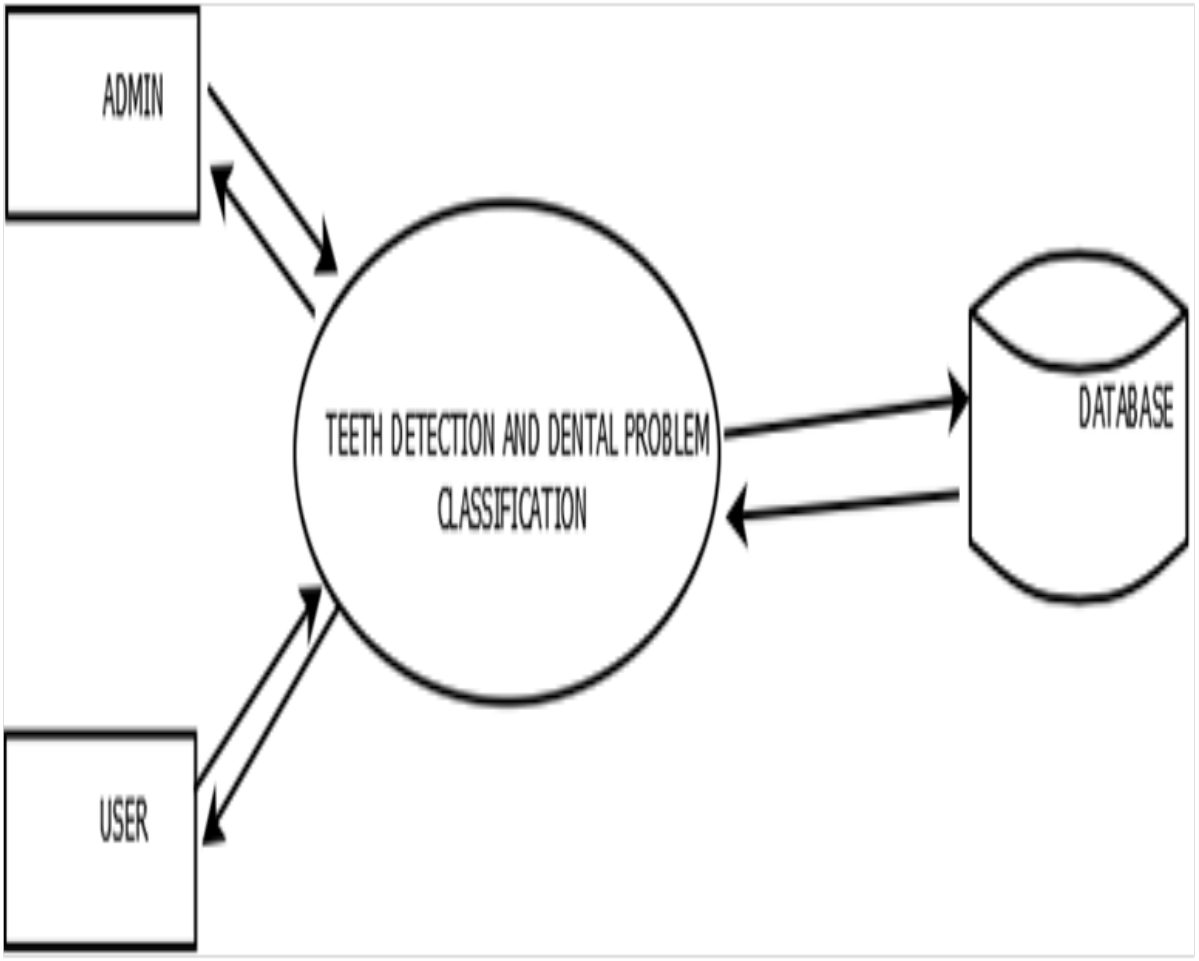

Figure A.1: level 0

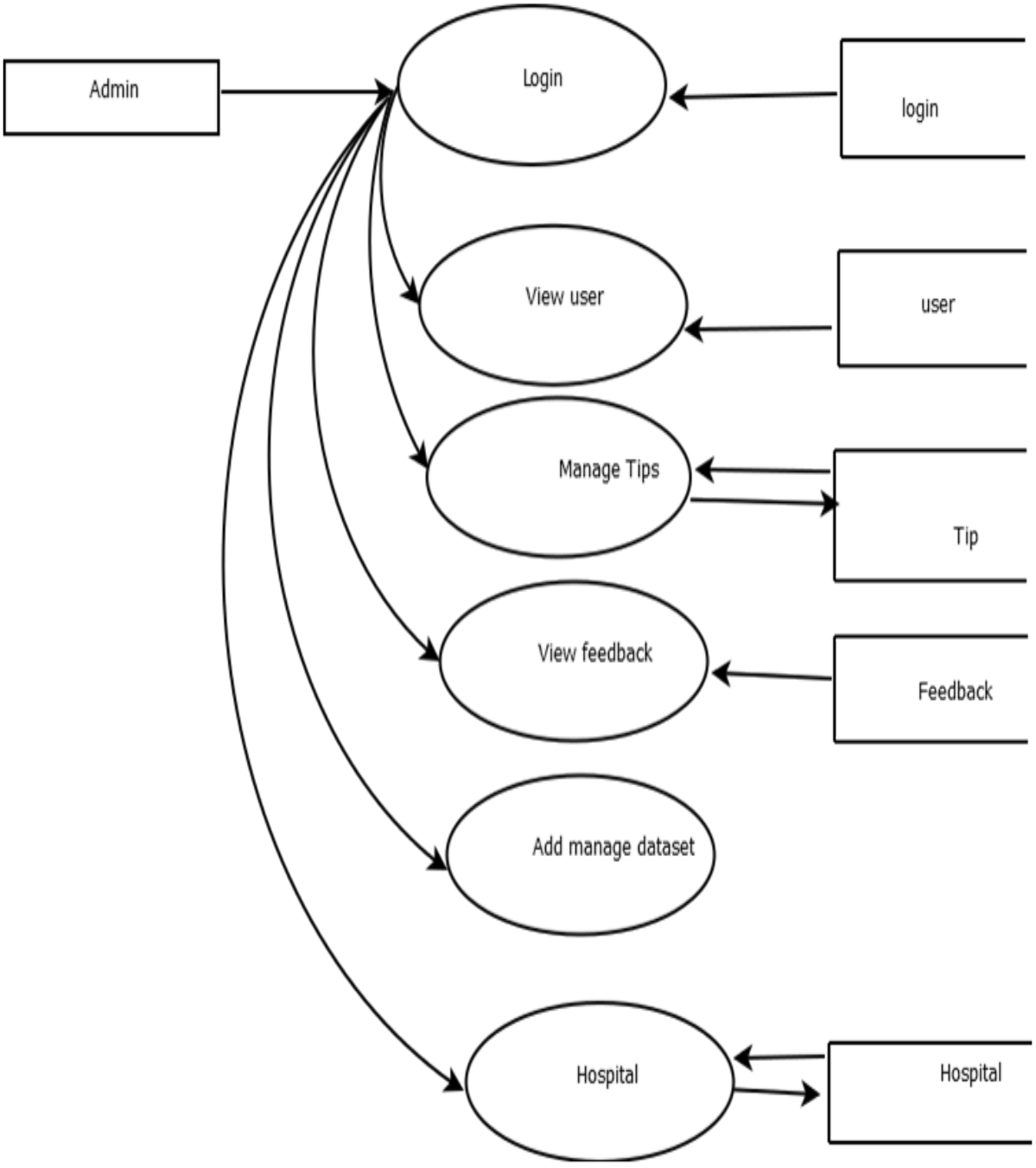

Figure A.2: level 1

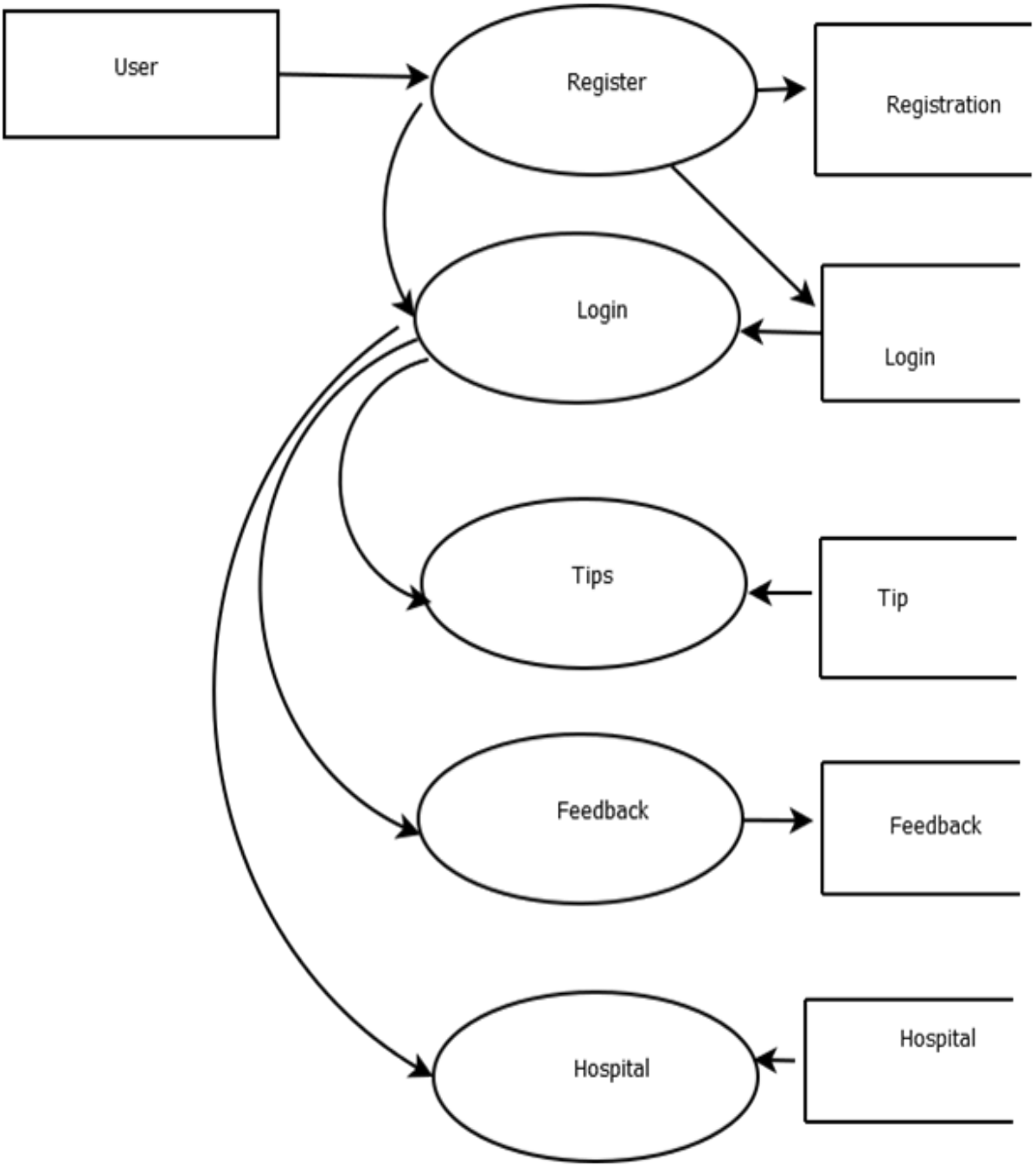

Figure A.3: level 2